

Python: module `esg.esgLs`

`esg.esgLs`

[index](#)

ESG file list markup.

Modules

[string](#)

Classes

[xml.sax.handler.ContentHandler](#)
[esgLsContentHandler](#)

class ***esgLsContentHandler***([xml.sax.handler.ContentHandler](#))

Methods defined here:

__init__(self)

setDocumentLocator(self, locator)

startElement(self, name, attrs)

startFile(self, attrs)

Methods inherited from [xml.sax.handler.ContentHandler](#):

characters(self, content)

Receive notification of character data.

The Parser will call this method to report each chunk of character data. SAX parsers may return all contiguous character data in a single chunk, or they may split it into several chunks; however, all of the characters in any single event must come from the same external entity so that the Locator provides useful information.

endDocument(self)

Receive notification of the end of a document.

The SAX parser will invoke this method only once, and it will be the last method invoked during the parse. The parser shall not invoke this method until it has either abandoned parsing

(because of an unrecoverable error) or reached the end of input.

endElement(self, name)

Signals the end of an element in non-namespace mode.

The name parameter contains the name of the element type, just as with the startElement event.

endElementNS(self, name, qname)

Signals the end of an element in namespace mode.

The name parameter contains the name of the element type, just as with the startElementNS event.

endPrefixMapping(self, prefix)

End the scope of a prefix-URI mapping.

See startPrefixMapping for details. This event will always occur after the corresponding endElement event, but the order of endPrefixMapping events is not otherwise guaranteed.

ignorableWhitespace(self, whitespace)

Receive notification of ignorable whitespace in element content.

Validating Parsers must use this method to report each chunk of ignorable whitespace (see the W3C XML 1.0 recommendation, section 2.10): non-validating parsers may also use this method if they are capable of parsing and using content models.

SAX parsers may return all contiguous whitespace in a single chunk, or they may split it into several chunks; however, all of the characters in any single event must come from the same external entity, so that the Locator provides useful information.

processingInstruction(self, target, data)

Receive notification of a processing instruction.

The Parser will invoke this method once for each processing instruction found: note that processing instructions may occur before or after the main document element.

A SAX parser should never report an XML declaration (XML 1.0, section 2.8) or a text declaration (XML 1.0, section 4.3.1) using this method.

skippedEntity(self, name)

Receive notification of a skipped entity.

The Parser will invoke this method once for each entity skipped. Non-validating processors may skip entities if they

have not seen the declarations (because, for example, the entity was declared in an external DTD subset). All processors may skip external entities, depending on the values of the <http://xml.org/sax/features/external-general-entities> and the <http://xml.org/sax/features/external-parameter-entities> properties.

startDocument(self)

Receive notification of the beginning of a document.

The SAX parser will invoke this method only once, before any other methods in this interface or in DTDHandler (except for setDocumentLocator).

startElementNS(self, name, qname, attrs)

Signals the start of an element in namespace mode.

The name parameter contains the name of the element type as a (uri, localname) tuple, the qname parameter the raw XML 1.0 name used in the source document, and the attrs parameter holds an instance of the Attributes class containing the attributes of the element.

The uri part of the name tuple is None for elements which have no namespace.

startPrefixMapping(self, prefix, uri)

Begin the scope of a prefix-URI Namespace mapping.

The information from this event is not necessary for normal Namespace processing: the SAX XML reader will automatically replace prefixes for element and attribute names when the <http://xml.org/sax/features/namespaces> feature is true (the default).

There are cases, however, when applications need to use prefixes in character data or in attribute values, where they cannot safely be expanded automatically; the start/endPrefixMapping event supplies the information to the application to expand prefixes in those contexts itself, if necessary.

Note that start/endPrefixMapping events are not guaranteed to be properly nested relative to each-other: all startPrefixMapping events will occur before the corresponding startElement event, and all endPrefixMapping events will occur after the corresponding endElement event, but their order is not guaranteed.



loadls(path)

Create a list of ESG File nodes from an XML file path.